The Debugging Task: Evaluating a Robotics Design Workshop

Emily Hamner, Tom Lauwers, Debra Bernstein

Carnegie Mellon University and the University of Pittsburgh
E. Hamner, CREATE Lab, Carnegie Mellon University, Pittsburgh PA
T. Lauwers, CREATE Lab, Carnegie Mellon University, Pittsburgh PA
D. Bernstein, UPCLOSE, University of Pittsburgh, Pittsburgh PA
etf@andrew.cmu.edu, tlauwers@andrew.cmu.edu, dlb36@pitt.edu

Abstract

Evaluating new educational programs and tools, especially those targeted at difficult-to-assess learning goals can be quite challenging due to the small number of participants typically engaged with pilot programs. The focus of the evaluation, then, should be on collecting rich data from each participant about their experience in the workshop and their progress towards meeting the workshop's learning goals. We present a novel evaluation technique, the debugging task, that seeks to assess at post-workshop a participant's independent ability to use the tools, skills, and materials of the workshop. The technique is presented in the context of Robot Diaries, a program to develop a robotics design activity centered on crafts materials and expressiveness, and targeted to middle school girls. The paper discusses the rationale for the debugging task, its implementation, and the results and analyses of girls completing the task.

Background

Evaluating new educational programs, tools, and interventions can be quite challenging. Typically, the difficulties stem from certain characteristics of new programs:

- A small number of participants are typically desired as untried interventions require a heavier teaching presence to catch problems as they appear.
- If the intervention targets learning goals that aren't typically addressed, a control group can not be established that spends an equal amount of time on a 'similar' educational activity.
- Some learning goals are much more difficult to measure than others. If the new intervention targets difficult to measure goals, this adds further complexity to the evaluation problem.

Essentially, these characteristics prevent the use of a number of standard evaluation techniques. The small number of participants and lack of a control ensure that there will not be enough power to show statistical significance. The learning goals themselves may be difficult to measure using traditional methods like exams.

This paper is about the creation and implementation of non-traditional methods for evaluation in the context of a program centered on improving technological fluency and design skills. We start with a description of the program, detail the learning goals, describe our general evaluation strategy and then detail the development and analysis of a novel, custom-tailored evaluation method for the program.

The Robot Diaries Program

The Robot Diaries program (Hamner et al 2008) was started to address the alarming gap in participation of women in computer science and engineering. As has been widely reported, engineering enrollments continue to drop throughout the United States (Vegso 2006). Even worse, there is increasing inequity; for instance women are very underrepresented in computer science and engineering, whereas science and business fields show significant improvement in female participation. Robotics has served as a popular vehicle for pipeline-based technology literacy programs because of its ability to attract and inspire the imagination of students who are often unmotivated by conventional classroom curricula (Druin and Hendler 2000). National contests including US FIRST, BEST, and Botball, programs have engaged more than 75,000 students (BEST Robotics 2009; Botball 2009; Melchior et al 2005). There is no doubt that some have found the contest-driven problem-solving experience to be transformative. However these existing robotics programs share a number of features that may limit participant diversity: they are short-term, high-intensity, competition-driven and technology focused.

In response, a number of researchers have proposed a complementary class of activities (Buechley 2007, Kim et al 2007; Resnick et al 2008) that we believe can engage and retain the participation of secondary level students who will not be attracted to the currently available interventions. Our program, Robot Diaries, aims to increase the technological fluency of our audience in order to significantly diversify the pipeline. By technological fluency, we mean the ability to manipulate technology creatively and for one's own use. We believe that our focus on fluency-building activities, which encourage creativity and personal adaptation of technology, engages a more diverse student population with technology and engineering.

In Robot Diaries students design affective, programmable tangible communication devices using familiar crafting materials and then use motors, lights and computation in novel ways to animate their creations in the context of emotional expression and collaborative storytelling. Robot Diaries is designed to motivate middle-school aged children to engage with technology as creators rather than as passive users., and to increase students' confidence in their ability to be creative with technology.

Over the course of the past three years, pilot Robot Diaries projects with 61 girls in six different Pittsburgh venues have enabled multiple rounds of participatory design with students and teachers as well as collection and analysis of the learning and self-confidence impact of this approach. The program's learning goals and approaches have evolved during this time; workshops in 2006 were primarily about determining the craft materials, software interface, and appropriate robotic components to use. Crucially, these workshops were taught by the research group. In 2007, the researchers reflected on the lessons of these pilots and developed a curriculum, new hardware, and new evaluation methods for pilots held in 2008. The learning goals and evaluation strategy described here are specific to a 2008 workshop held with a seven-student homeschool group. The workshop was taught by parents and observed by researchers. It consisted of six three- to four- hour sessions held over the course of seven weeks.

Evaluating Learning Goals

At the outset of our 2008 redesign of the curriculum and technology we identified the learning goals of the Robot Diaries program. These goals were used to guide our instructional and evaluation strategy and are detailed here.

Learning Goals

The ultimate goal of Robot Diaries is to enable girls to engage with, change, customize or otherwise become fluent with the technology in their lives. In order to achieve this goal, we are engaging in a two-pronged strategy. First, we address girls' *dispositions towards technology.* Second, we provide girls with the *knowledge and skills* they need to engage fluently with technology (NRC 1999).

The dispositional goals of the program are to help girls see technology as interesting and deeply relevant to their lives, to help motivate their continued engagement and exploration of technology, and to provide them with confidence in their own ability to create with, modify, or troubleshoot the technology in their lives.

The central knowledge and skills goals of Robot Diaries are focused on design and creation; although some of these goals are specific to the Robot Diaries technological context, many of them are in line with the ITEA Standards for Technological Literacy (ITEA 2002). With respect to design, girls will understand and be able to engage in an iterative design process, including prototyping, evaluating, troubleshooting, and documenting. They will understand the idea of trade-offs and constraints and be able to identify them for specific designs they create. In terms of creation, girls will receive a detailed understanding of the robotic and structural kit components to allow them to properly identify components which meet their needs. Girls will also learn to use a number of tools to construct their designs, as well as a graphical programming language to animate their creations.

Evaluation Strategy

As the number of participants was very small (seven) and this was the first workshop offered with our new curriculum and hardware, our evaluation strategy centered on collecting a range of information from each girl. We employed a number of fairly standard methods, notably:

- **Pre/Post Interviews.** Each girl was interviewed before the workshop began and after it ended. Interviews directly asked girls about their interest in robotics, science and technology, and attempted to detect change in our dispositional learning goals. Interviews also included questions about relevant declarative knowledge (e.g., identify and provide a definition for relevant parts, such as sensors and servos) and designed systems (e.g., examine an electronic toy and describe how it works).
- **Research Observations.** One or more researchers attended every workshop session, logging girls' reactions to new activities, documenting moments of frustration or achievement, and determining how closely the parent instructors followed the curriculum. The workshops were also video recorded to allow the entire research team to view moments of specific interest.
- **Instructor Interviews.** We interviewed instructors to get their impressions on how the girls proceeded through the program, as well as gather information about what was difficult in implementing the curriculum.
- **Parent Interviews.** We interviewed parents at the beginning and end of the workshop. Interviews focused on the child's previous experience with robotics and related technologies, the family's activities related to science and technology, and parents' impressions of the workshop and what their child gained from participation.

These methods were sufficient to provide us with information relating to the dispositional learning goals – we received data regarding girls' dispositions towards science and technology from the girls themselves, from our own observations, and from the instructors and parents. They also provided some information related to the knowledge goals. However, we also wanted to assess girls' change in creativity and design skills as well as technical skills such as debugging. While the instructor interviews and our own observations provided some subjective information on this, we felt it necessary to create two new methods that explored change in these learning goals in a less subjective and finer grained way. These methods were:

- **Debugging Task.** To assess whether girls were able to work with and troubleshoot the hardware and software after the workshop concluded, we gave them a robot made from the same materials and components that was broken in five distinct ways and asked them to fix it.
- **Creative Design Exercise.** To assess change in how girls thought about designing and attempt to measure transfer of design ability, at least within the context of expressive electronic devices, we asked them both before the workshop started and after it concluded to sketch and model a design of a device that could solve a community problem.

The remainder of this paper focuses on the design, implementation, and analysis of the debugging task. A second paper will be released in the future to discuss the Creative Design Exercise.

The Debugging Task

Task Structure

Each participating girl was shown a video of a craft robot performing a four-step program. On the table in front of her was that same craft robot, a laptop with the programming software displaying a sample program, a selection of tools and replacement parts, and a reference handout from the workshop. She was given 25 minutes to make whatever modifications were necessary, either to the robot or to the program, in order to make the robot imitate the video. The starting program, tools, spare parts and reference sheet were all pointed out to the participant and she was shown how to play and pause the video.

The robot is programmed by moving sliders which directly control the servos, LEDs, etc; the interface is shown in figure 1. When the components are in the desired state, one saves an “expression”. The state of multiple components can be saved in to a single expression thus allowing, for example, the robot to both move a servo and turn on an LED at the same time. On the other hand, not all components need be part of an expression; if an expression is saved while only LED sliders are open, it will have no effect on the position of the servos. Multiple saved expressions can then be linked together to form a simple sequential program, referred to as a “sequence”.

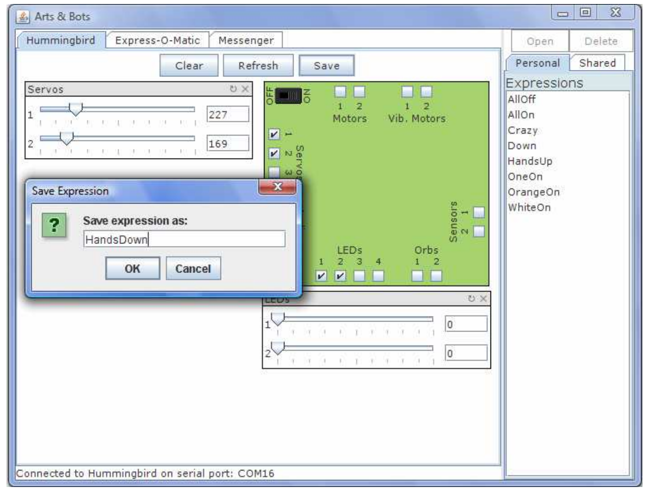

Figure 1: The programming software. Sliders are used to set the state of servos, LEDs, and other robot components. The set of state parameters are saved as an “expression” which can later be linked together with other expressions to form a simple sequential program.

The robot contained the following parts: 4 LEDs, 2 servo motors, 1 vibration motor, and a Hummingbird microcontroller. The goal of the program design was for each step to focus on a different body part: the ears, the eyes, the bell, the nose (Fig. 2). The video began by showing the robot with all lights off and the ears down (some girls treated this as a step although we had not intended it to be). In an attempt to demarcate the individual steps, the ears moved with each step: up, down, up, down. The eyes turned off at step 2 and back on for the remainder of the program. In step 3, a motor began to vibrate causing a bell to ring. In step 4 the LED in the nose turned on. Finally, in step 5, all lights and motors turned off. The sample software program contained five steps named ‘A’, ‘B’, ‘C’, ‘D’, and ‘Stop’.

Prior to the start of the task, five intentional problems (bugs) had been introduced to the robot and sample program (listed in order of increasing difficulty):

- [hardware] the Hummingbird microcontroller was turned off
- [hardware] the two LEDs in the eyes were unplugged from their Hummingbird ports
- [software or hardware] the vibration motor was plugged into a different Hummingbird port than was indicated in the sample program
- [software] one of the steps in the program was incorrect (step 2 did not turn off the LEDs in the eyes)
- [hardware] the LED in the robot’s nose was not functional

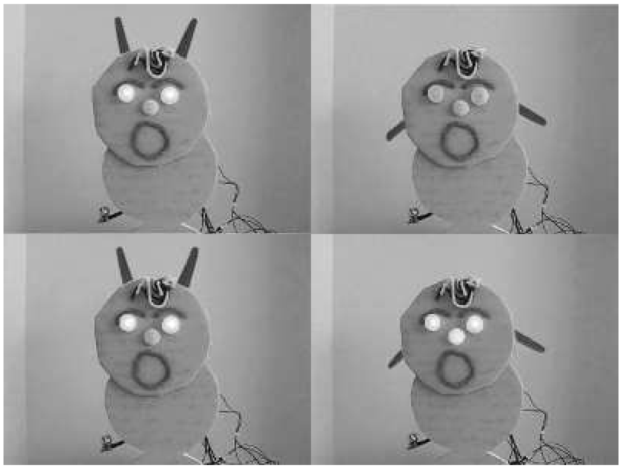

Figure 2: Steps in the target program. *Top left:* Step 1 - ears up, LEDs in eyes turn on. *Top right:* Step 2 - ears down, LEDs in eyes turn off. *Bottom left:* Step 3 - ears up, LEDs in eyes turn on, vibration motor rings bell. *Bottom right:* Step 4 - ears down, LEDs in eyes stay on, vibration motor continues to ring bell, LED in nose turns on.

The task was designed to test overall debugging skills while each item from the list was designed to test a specific skill necessary to use the hardware and software, such as connecting components to the microcontroller, understanding the building blocks of programs, and putting steps together in to a sequential program.

Each girl in the workshop participated in the task individually and was asked not to share her solutions with girls who had not yet completed the challenge. The researcher administering the challenge observed the student as she worked, took notes, and encouraged her to speak her thoughts as she worked on the challenge; the setup is pictured in figure 3. If asked a question the researcher tried to deflect the question by saying things such as "what do you think?" rather than providing solutions to the challenge. The student was recorded by a video camera and audio recorder as she attempted to complete the task. When available, a second video camera recorded a close-up of the laptop screen. Upon the conclusion of the challenge, the participant was asked a few questions about the difficulties she encountered and her strategy and choices debugging the robot.

Data Analysis

Our analysis approach was three-fold. First, we sought to document *whether* participants were able to successfully resolve the 5 bugs listed above. Second, we sought to document *how* participants resolved the bugs (i.e., what strategies they used to do so). Third, we analyzed the *errors* participants made while attempting to debug the robot.

Figure 3: The debugging task setup with a student working on the challenge as a researcher observes.

We took several steps in order to analyze the data from the challenge. First the video of the challenge was transcribed. Two or 3 researchers then watched each video while reviewing and discussing the transcript. (One researcher came from a background in learning research and the other two from a background in robotics.) To help in categorizing the debugging choices made by the students, the researchers made a list of steps that the roboticists considered logical, quasi-logical or illogical when faced with each bug. The transcript was broken into sections based on which of the 5 bugs the participant was working on. When she resolved a bug, the time was noted. During the review of the video and transcript, researchers noted any behaviors that stood out as particularly insightful or particularly misguided. A short discussion followed each review in which researchers looked for common themes in the strategies employed or mistakes made by the participant.

Sample Results

Three girls successfully solved every bug. All seven of the girls resolved the simplest three bugs (Hummingbird was off, LEDs were unplugged, motor was in the wrong port). Four out of 7 corrected the programming, and 5 out of 7 replaced the burned out LED.

While the method described above took a significant amount of time, it yielded several insights in to the knowledge and understanding of the students. In particular the struggles of the 3 girls who did not completely solve the task highlight some of the areas where future curriculum or software improvements could be made. The following four accounts of girls solving the debugging task describe the types of information and insights the research team garnered from analyzing the task.

Dakota

The oldest participant in the workshop, Dakota[1], was 14 years, 1 month old at the start of the workshop. Our workshop observations indicated that she had a strong understanding of the robot and programming environment. In the debugging challenge, she showed the research team that she had developed a stronger set of robotics skills than we expected purely from our workshop observations.

Dakota was able to successfully solve the challenge in approximately 14 minutes. We noted in our analysis conversations her systematic approach to debugging and that she proceeded from most likely to least likely to fail when tracking down a bug (i.e. she started by checking the software, a common source of failure, and only at the end moved on to the least likely aspect to fail – the hardware itself). We also noted that Dakota went 'above and beyond' the expected debugging skills by plugging the faulty LED into another port and confirming that it was not functional before substituting a new LED. This was a particularly impressive step on Dakota's part, one the group would not necessarily expect to see someone perform.

Noel

At 9 years, 7 months, Noel was the youngest child in the workshop. During the workshop itself, Noel often needed additional help from the instructors. In our analysis of the debugging challenge, we identified two specific problems for Noel which were not readily apparent from our observations during the workshop.

First, we believe she failed to make a connection between the 3 vital components on the robot – the controller/Hummingbird, the robotic components (e.g., LEDs), and the software. This problem was evident when she tried to fix the nose LED; she repeatedly tried to control the nose LED which was plugged in to port 4 by moving LED slider 3. Even after examining the LED wires, unplugging them and plugging them back in, she continued to try to operate the LED using the incorrect slider. Her failure to recognize the connection between all 3 of these components made it difficult for her to observe the real problem with the nose. Even after she replaced the faulty hardware, she was not able to make the LED light up with her program, despite observing that it functioned when running the sample program.

We believe Noel's second difficulty was that she did not fully understand the hierarchical nature of programming. We base this conclusion on several curious choices she made as she programmed the robot. One such curiosity was her choice to position the robot correctly (using the sliders) for some steps in the sequence but then failing to save the positions as an expression before moving on to the next step. Another is her choice to make multiple expressions in order to accomplish what is a single step in the video (she did this for the step with the vibration motor). We also noticed that she did not tend to re-use expressions. We believe all of these choices are consistent with her not fully understanding the hierarchical nature of programming.

Allison

Allison was 10 years, 10 months old at the start of the workshop. From our workshop observations and interviews with Allison, we knew that she did not have much computer experience, though she enjoyed programming her robot. Consistent with those observations, we found that Allison's biggest problem in solving this task was with the programming.

Over the course of the task, she made 3 attempts at putting together a sequence for the robot, none of which were successful. Her first attempt (using 4 steps she created and the final 'Stop' step from the sample program) matched the video for the start state, step 3, step 4, and the stop state. Her second attempt yielded a sequence that was consistent with the video for steps 1 and 2, and possibly 3. She was not able to finish her third programming attempt, but she began to use a new strategy of writing steps down on paper during this programming attempt. Interestingly our analysis indicates that Allison actually had all of the expressions she needed to successfully program the robot, but she was unable to put the steps together into a single sequence to make the correct program. We believe that her approach of writing things down might have been promising for her if she had had more time.

Two problems surfaced during her programming attempts. One is that she sometimes saved expressions with the robot's parts in the physically correct positions but the program sliders closed (so that the positions, while appearing correct, did not save as part of the expression). She also had a lot of difficulty getting the correct number and sequence of steps in the program. The fact that one or two of the expressions did not actually match steps in the video only added to her confusion.

Tracy

Tracy, age 11 years 0 months, was a successful and proficient programmer in the context of the workshop. She frequently created programs at home during the week in between workshop sessions. We were thus somewhat surprised to discover a few gaps in her programming knowledge that could have strengthened her programming ability.

Tracy came close to successfully re-programming the robot, but her final sequence was not quite correct; the eyes flashed an extra time in the middle of the program, the bell came on too early, and the nose light came on a bit too early. (Tracy herself acknowledged the problems of timing and the bell coming on too early). Mostly these were issues of timing, which might be endemic to her general approach of having a large number of expressions in her sequence. Instead of programming steps that had multiple elements engaged at once, she mostly had a one-element-at-a-time approach to the programming. In reviewing her expressions, we noted that she did sometimes have sliders open for multiple elements, but never were multiple

[1] Names have been changed.

elements turned *on* in a single expression. The result of this was that she often used multiple steps to do what a single step accomplished in the video.

Tracy encountered a few other specific difficulties. One is that we don't think she understood how to look inside the content of a completed expression (i.e., that she could click on an expression from the list of saved expressions in order to open and examine it). A related problem was her 'all off' expression. Several times she seemed confused as to why this expression didn't shut the vibration motor off, and in response to questioning at the end of the challenge, she was unable to explain why the 'all off' expression didn't stop the motor. There is a simple knowledge element here; an expression will only control the elements for which sliders are *open*. Since her 'all off' expression only had the LED sliders open, it didn't control the vibration motor. Knowing that she could open and review an expression might have helped Tracy to recognize that the vibration motor was not in fact controlled by her 'all off' expression.

Discussion and Conclusions

The debugging activity analyses provided the research team with a number of insights into girls' progress towards the knowledge and skills learning goals. Based on the difficulties faced by the students, we were able to identify areas of improvement for our instruction and software interface that, when implemented in future workshops, will likely improve the knowledge and skills outcomes. Some examples of improvements stemming directly from these analyses are:

- The curriculum could include a brief second software instruction session highlighting the areas that gave students trouble. Specifically additional instruction reminding students that only the open sliders are saved in an expression might help reduce errors. Demonstrating this with a short sample program might help students visualize how this plays out in a sequence.
- Students should be reminded that the state of all open sliders will be saved in the expression, thus multiple components can be controlled by a single expression.
- Students might benefit from explicit instruction on how to review which components are impacted by a particular expression. While some students had no problem with this, others could have benefited from additional instruction on how to open and view a saved expression.
- Software improvements could be made to mitigate interface specific errors like forgetting to turn on the slider of an element that needs to be controlled.

While these suggestions are specific to our program, we feel that the implementation of a debugging task as evaluation tool is applicable to a wide variety of creative design activities in which the number of participants is small and learning goals are difficult to measure. The richness of the information garnered through the task was well worth the time required to design and analyze the task.

References

BEST Robotics 2009: http://www.bestinc.org

Botball 2009: http://botball.org

Buechley, L. and Eisenberg, M. (2007) Fabric PCBs, Electronic Sequins, and Socket Buttons: Techniques for E-textile Craft. Journal of Personal and Ubiquitous Computing, June 2007.

Druin, A. and Hendler, J. 2000. Robots for kids: exploring new technologies for learning. The Morgan Kaufmann Series in Interactive Technologies, Morgan Kaufmann.

Hamner, E., Lauwers, T., Bernstein, B., Nourbakhsh, I., and DiSalvo, D. Robot Diaries: Broadening Participation in the Computer Science Pipeline through Social Technical Exploration. Proceedings of the AAAI Spring Symposium on Using AI to Motivate Greater Participation in Computer Science, p.38-43, March 23-26, 2008, Palo Alto, California, United States.

International Technology Education Association (ITEA) (2002). Standards for technological literacy: Content for the study of technology. Reston, VA: International Technology Education Association.

Kim, H., Coluntino, D., Martin, F., Silka, L., Yanco, H., "Artbotics: Community-based collaborative art and technology education." In Proceedings of SIGGRAPH 2007, The 34th International Conference and Exhibition on Computer Graphics and Interactive Techniques, San Diego, CA, August 2007.

Melchior, A., Cohen, F., Cutter, T., and Leavitt, T., "More than Robots: An Evaluation of the FIRST Robotics Competition Participant and Institutional Impacts", Heller School for Social Policy and Management, Brandeis University, 2005.

National Research Council (NRC). (1999). Being fluent with information technology. Washington, DC: National Academy Press.

Resnick M. Berg R. Rusk, N. and M. Pezalla-Granlund. New pathways into robotics: Strategies for broadening participation. Journal of Science Education and Technology, Volume 17, Issue 1, pp.59-69 , 2008.

Vegso, J. 2006. "Drop in CS Bachelor's Degree Production." Computing Research News, Vol. 18 (2), 2006.